Pulse Stretcher VME Module

The Extraction Kicker Application

User Manual

Craig Drennan
September 26, 2008

I. Introduction

The Pulse Stretcher VME Module was originally developed for monitoring Booster extraction kicker current pulses. These pulses are only a couple microseconds in duration and it was desired to expand these signal in time so they could be sampled with the standard ACNET MADC digitizers. The MADC digitizers already have all of the plotting and data logging support applications available. The Pulse Stretcher modules use a digital signal processing approach of digitizing the analog signal, processing the signal within an FPGA, and re-generating the new analog signal through a digital to analog converter (DAC). This approach not only provides a method for re-timing or stretching the signal, but tests and signal threshold comparisons are made also.

An alternative to re-digitizing the signals with the MADC system would be to digitize the signals and transfer the signal data across the VME bus to a processor module that would then serve the data to the ACNET plotting routines and data logging. This approach would require the services of more programmers. For a small number of instrument applications the Pulse Stretcher's digital signal processing of analog signals approach can be an efficient use of manpower and provide flexibility in developing application specifications.

The down side of digitizing and regenerating the analog signals is that there is always the introduction of additional noise and distortion. However, in our current application and several others, the additional noise and distortion is not significant. Effort to calibrate all of the ADC's and DAC's has been made for each module. The gain and offset adjusting coefficients for each module are stored in non-volatile EEPROM memory. These settings are made over the VME interface at the test bench.

A second disadvantage to this approach is that the number of cables and connections for the analog signals double. This could be unacceptable in applications that monitor even a moderate number of signals in a particular location. The current application has only four signals at four locations around the Booster, so the additional number of cables is not unacceptable.

Inquiries by electronic support groups here at the lab for supporting the type of measurements and processing used in this application are welcome. Contact Craig Drennan at cdrennan@fnal.gov if you are interested.

II. Booster Extraction Kicker Signal Monitoring

The goal is to detect sparking or arcing of the Booster extraction magnets. Bdot coils are mounted to both the power supply and power return leads of the magnets in the Booster enclosure. In normal operations equal (and perhaps opposite depending of coil orientation) signals will be produced by each coil. By monitoring the difference (or sum) of these two Bdot signal we can detect the arcing of power to ground.

The overall configuration of the electronics is shown in Figure II.1. The module uses three ADC inputs and three DAC outputs to produce the stretched signals that can be plotted through the MADC channels. The fourth channel's ADC input is used to input a Threshold value, which is described further, later in the paper. The fourth channel's DAC output is used to generate the signal representing the difference between the two Bdot signals. One TTL input to the module is used as a "Start" trigger input. One TTL output is used as an indication that the difference threshold was exceeded.

The input signals are stretched in time by a factor of 12.5. The inputs are sampled at a rate of 250 k samples per second. These samples are then output by the DAC to regenerate the analog signal at an update rate of 20 k samples per second. The input is sampled 250 times over a 1 milli-second interval. These 250 samples are played back out over a 12.5 milli-second interval. All this is illustrated in Figure II.2.

A DC input voltage on the fourth ADC channel of the module is used as a threshold. If the difference between the Bdot signals exceed the magnitude of this threshold, in either the positive or negative direction, the TTL Spark Detection signal will latch "High" until the start of the next triggered data collection interval. In order to avoid the normal transient swings in the Bdot signals a pair of settable parameters are used to define the interval in which the threshold comparison is to take place. Additionally, there is another parameter which declares a number of samples at the beginning of the record to be averaged into a baseline for the difference measurement. Figure II.3 further defines the parameters that define the comparison and baseline intervals.

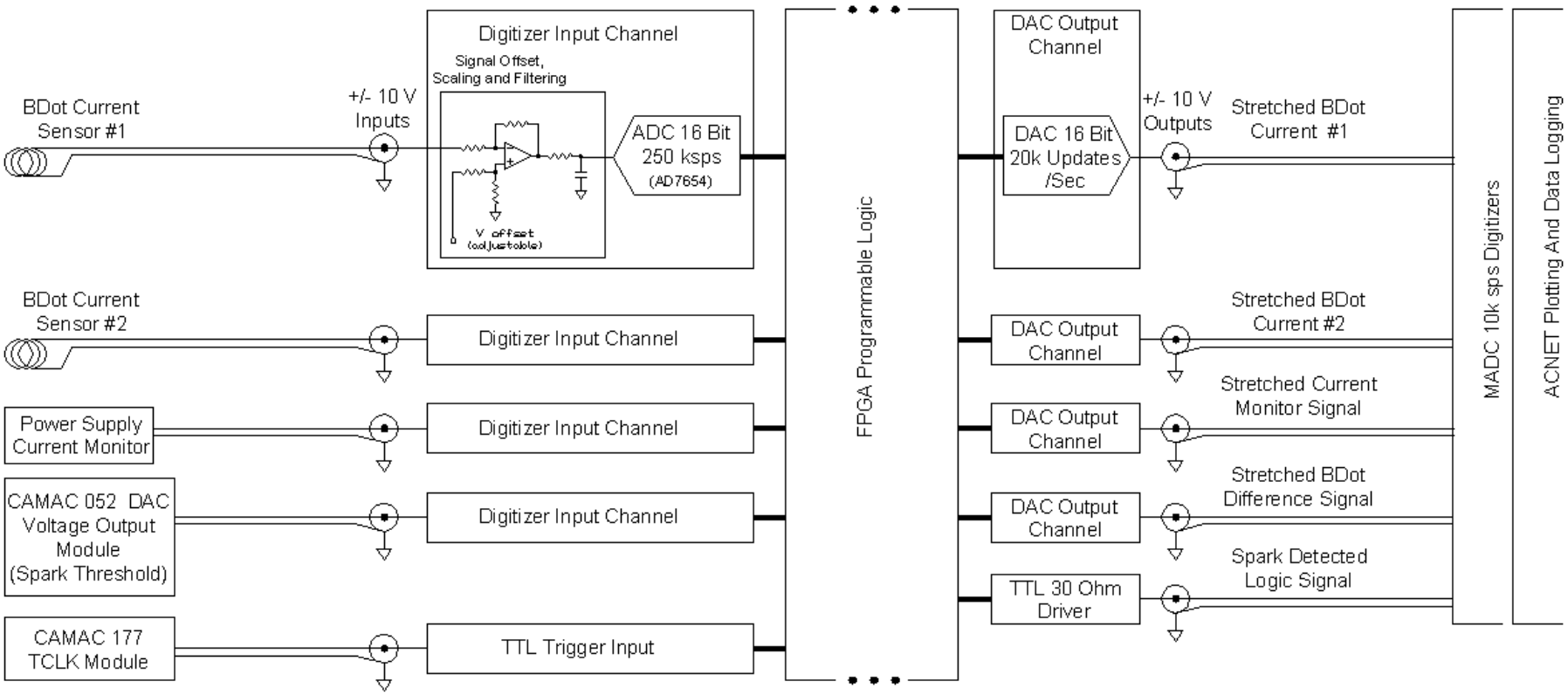


Figure II.1 Block Diagram of the Booster Extraction Magnet Application.

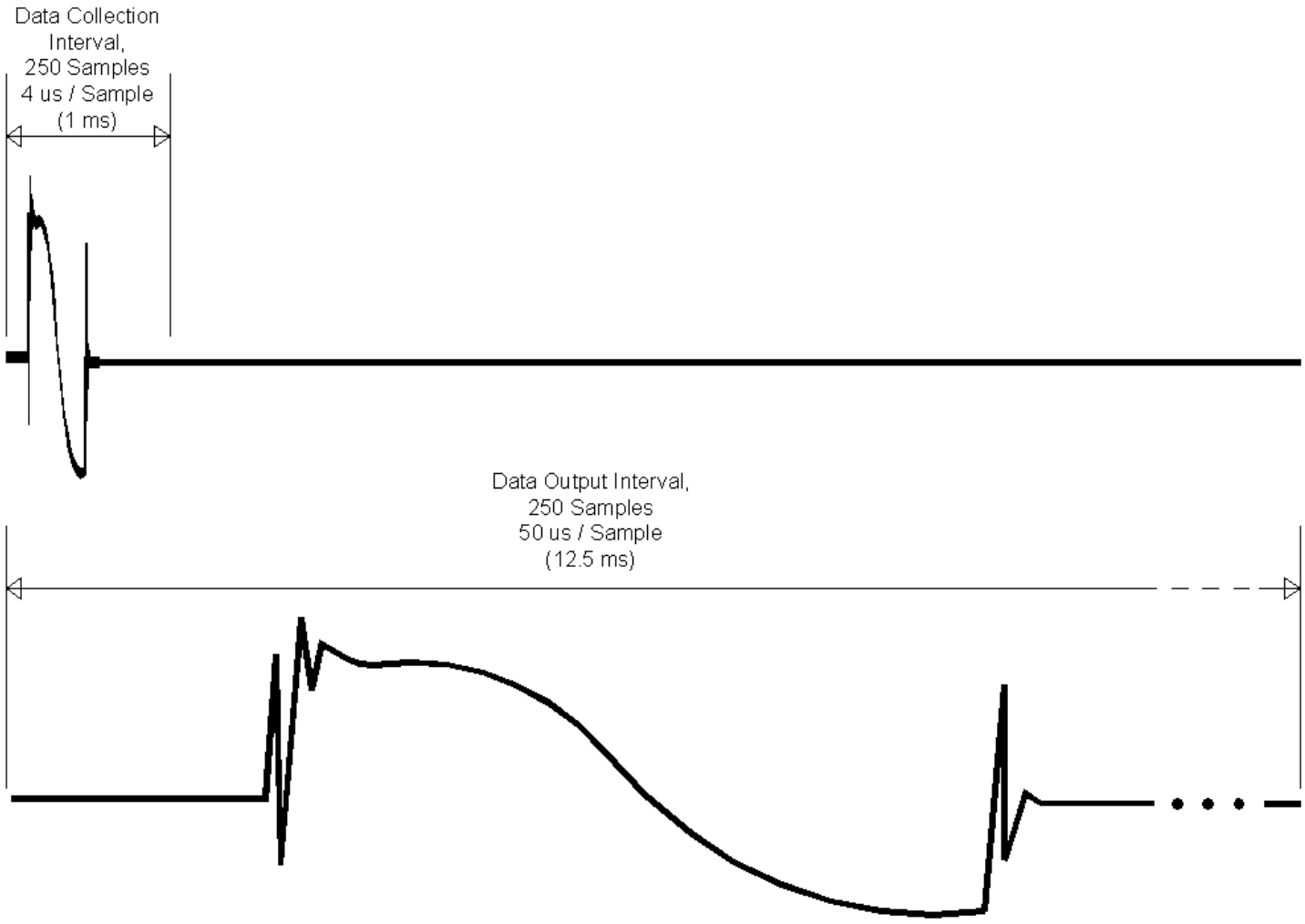


Figure II.2 Illustration of the Signal Expansion in Time.

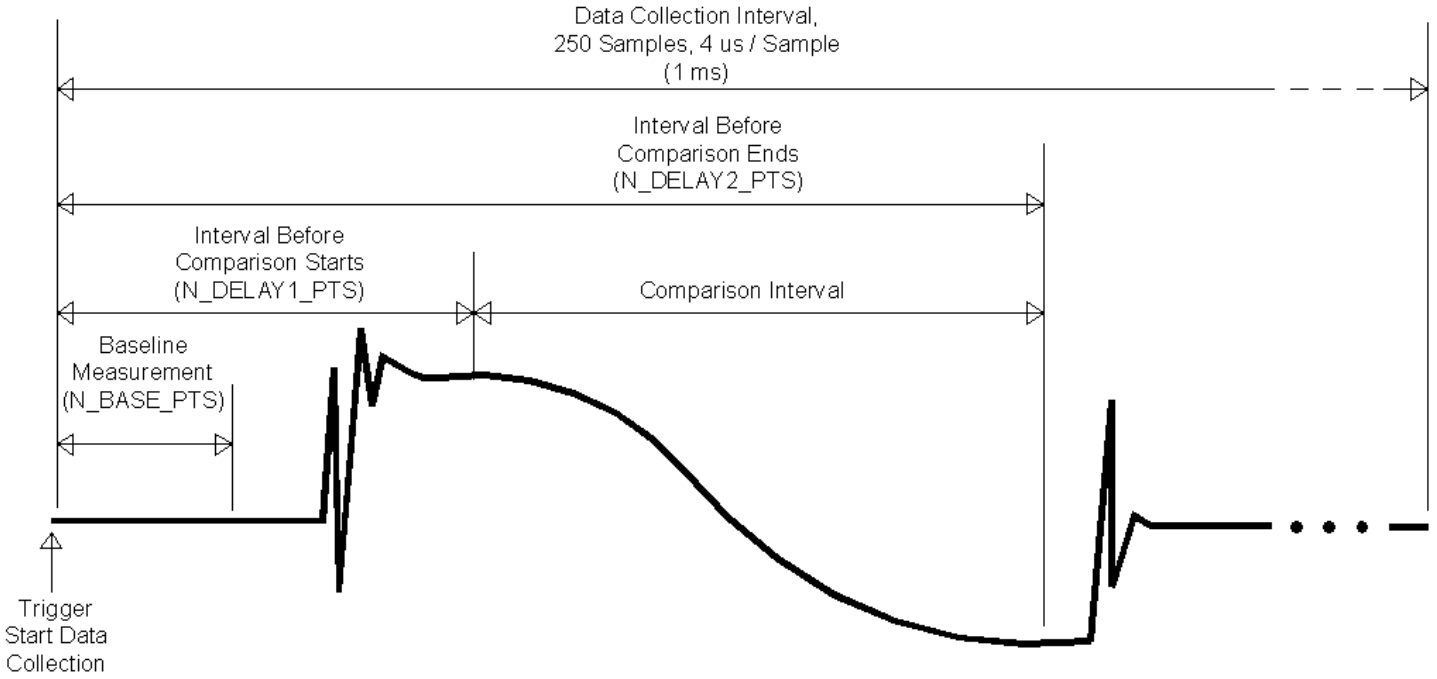


Figure II.3 Definition of the Parameters That Define the Comparison and Baseline Intervals.

III. Implementation Details

The implementation details here include the specific ACNET connections and device names, and the VME bus memory map for accessing the settable parameters in the module. The ACNET connections and devices are given below. The VME memory map is given in Table III.1.

Signal Connections for MP02

Comparator Threshold Reference Voltage:

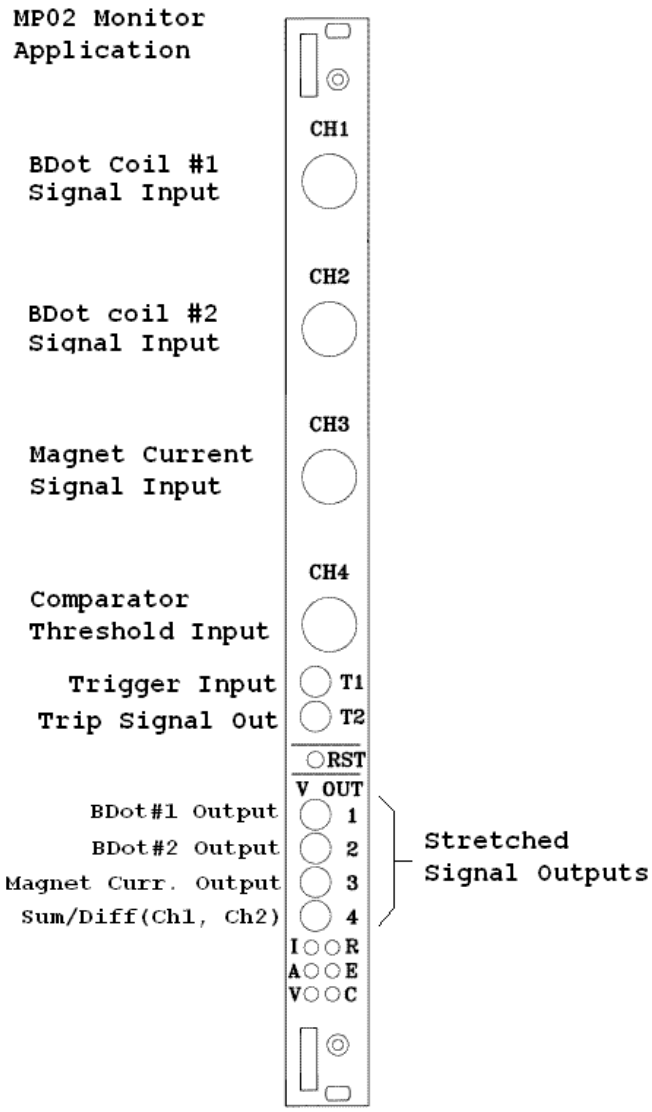
B: TLVL02
CAMAC 052 DAC Voltage Output Module
Crate: $42
Slot: 9
Channel: DAC2

"Start" Trigger Input:

B: TDAQ02
CAMAC 177 Tclk Module
Crate: $41
Slot: 5
Channel: 4

Analog Signals Out:

MADC#11
Crate: $40
Slot: 20



MADC Channel:

24 B:BDQT1 "Bdot Coil #1 stretched signal, MP02 (Bdot#1 Output)"

25 B:BDQT2 "Bdot Coil #2 stretched signal, MP02 (Bdot#2 Output)"

26 B:IMP02 "MP02 Load Current (Magnet Curr. Output)"

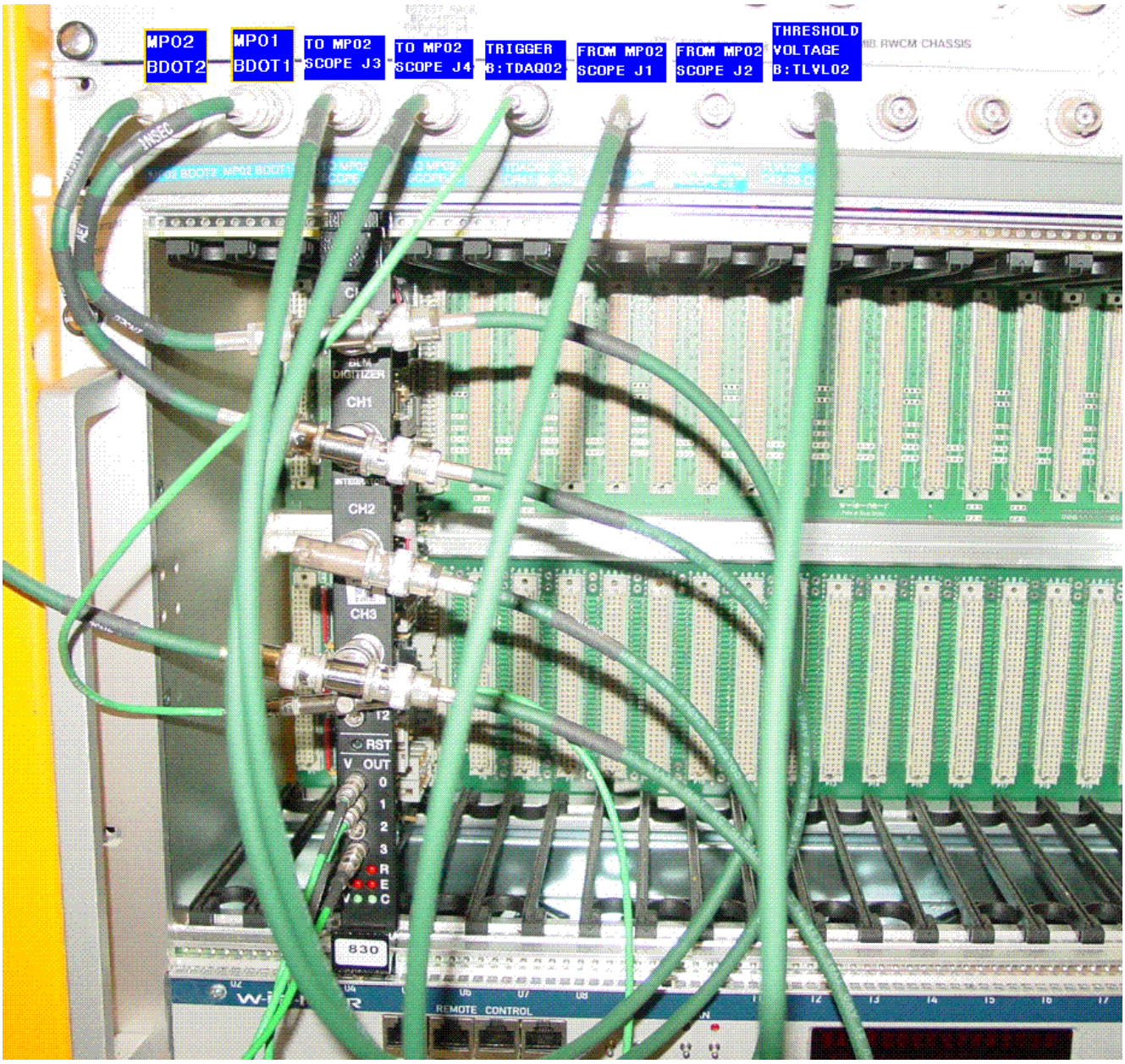
27 B:DBDT12 "Difference Between Bdot #1 and Bdot #2, MP02 (Sum/Diff)"

28 B:BDTTRP "Difference beyond Threshold (sparking), MP02 (Trip Signal Out)"

Table III.1 Pulse Stretcher VME Memory Map

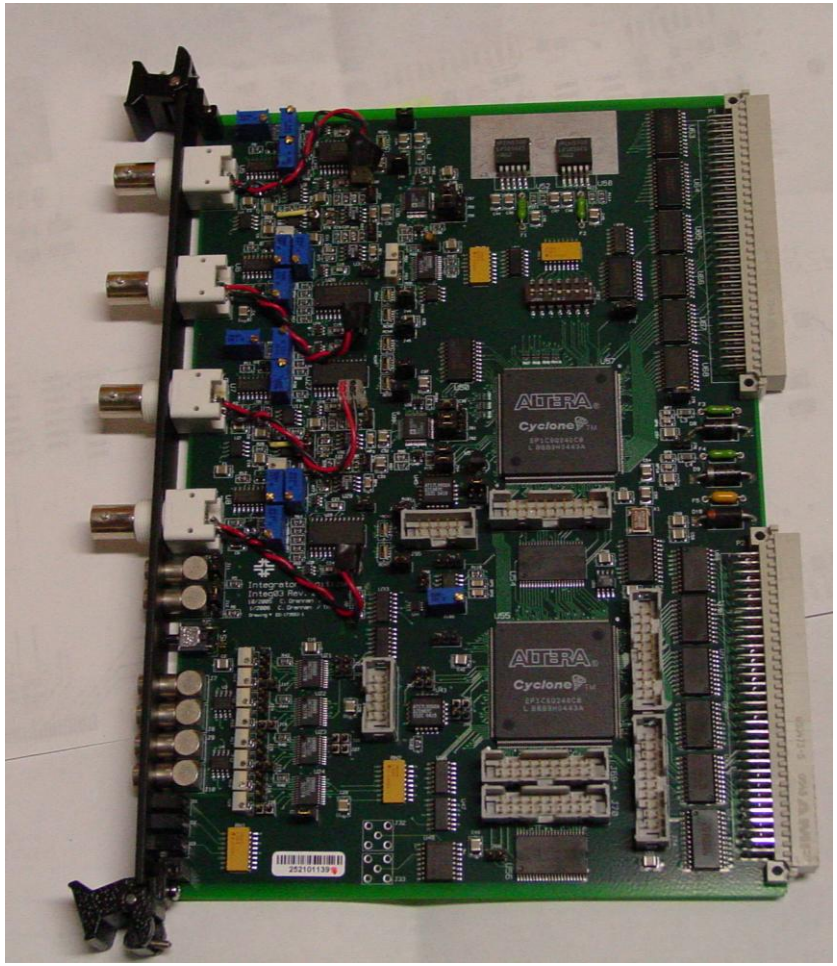
Note: N in the addresses stand for the Hex value of the board address 0-E set by the DIP switches on the board.

Register / Command / Memory	**Address [19..0]**	**R/W**	**Description**
Board ID Memory (ROM)	0xN00000 to 0xN001FE	R	Board ID number and text string. (1 character per 16 Bit word.)
Board Serial Number	0xN00200 to 0xN00206	R	Board Serial Number from the DS2401 ROM. device, 64 Bits.
Channel 1 ADC Gain Correction	0xN03000	R/W	Channel 1 ADC Gain Correction
Channel 1 ADC Offset Correction	0xN03002	R/W	Channel 1 ADC Offset Correction
Channel 2 ADC Gain Correction	0xN03004	R/W	Channel 2 ADC Gain Correction
Channel 2 ADC Offset Correction	0xN03006	R/W	Channel 2 ADC Offset Correction
Channel 3 ADC Gain Correction	0xN03008	R/W	Channel 3 ADC Gain Correction
Channel 3 ADC Offset Correction	0xN0300A	R/W	Channel 3 ADC Offset Correction
Channel 4 ADC Gain Correction	0xN0300C	R/W	Channel 4 ADC Gain Correction
Channel 4 ADC Offset Correction	0xN0300E	R/W	Channel 4 ADC Offset Correction
Channel 1 DAC Gain Correction	0xN03010	R/W	Channel 1 ADC Gain Correction
Channel 1 DAC Offset Correction	0xN03012	R/W	Channel 1 DAC Offset Correction
Channel 2 DAC Gain Correction	0xN03014	R/W	Channel 2 DAC Gain Correction
Channel 2 DAC Offset Correction	0xN03016	R/W	Channel 2 DAC Offset Correction
Channel 3 DAC Gain Correction	0xN03018	R/W	Channel 3 DAC Gain Correction
Channel 3 DAC Offset Correction	0xN0301A	R/W	Channel 3 DAC Offset Correction
Channel 4 DAC Gain Correction	0xN0301C	R/W	Channel 4 DAC Gain Correction
Channel 4 DAC Offset Correction	0xN0301E	R/W	Channel 4 DAC Offset Correction
N_BASE_PTS	0xN03020	R/W	Number of samples to average to compute the difference baseline.
N_DELAY1_PTS	0xN03020	R/W	Number of samples before the start of the compare interval.
N_DELAY2_PTS	0xN03020	R/W	Number of samples until the end of the compare interval.


MP02
BDOT2
MP01
BDOT1
TO MP02
SCOPE J3
TO MP02
SCOPE J4
TRIGGER
B:TDAQ02
FROM MP02
SCOPE J1
FROM MP02
SCOPE J2
THRESHOLD
VOLTAGE
B:TLVL02
CH1
CH2
CH3
RST
V OUT
0
1
2
3
R
E
C
830
REMOTE CONTROL

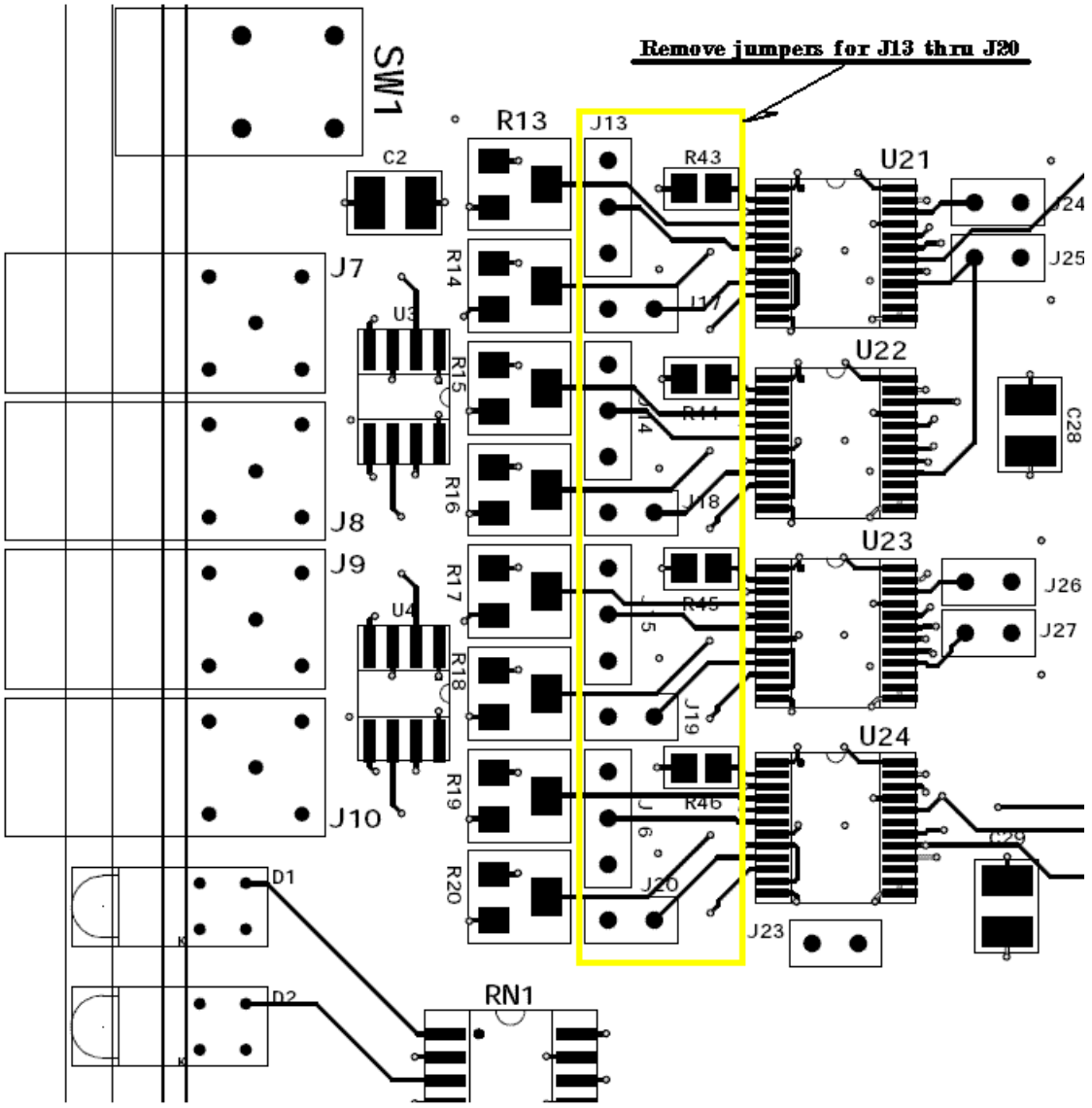
APPENDIX A

Converting to the pulse stretcher application.



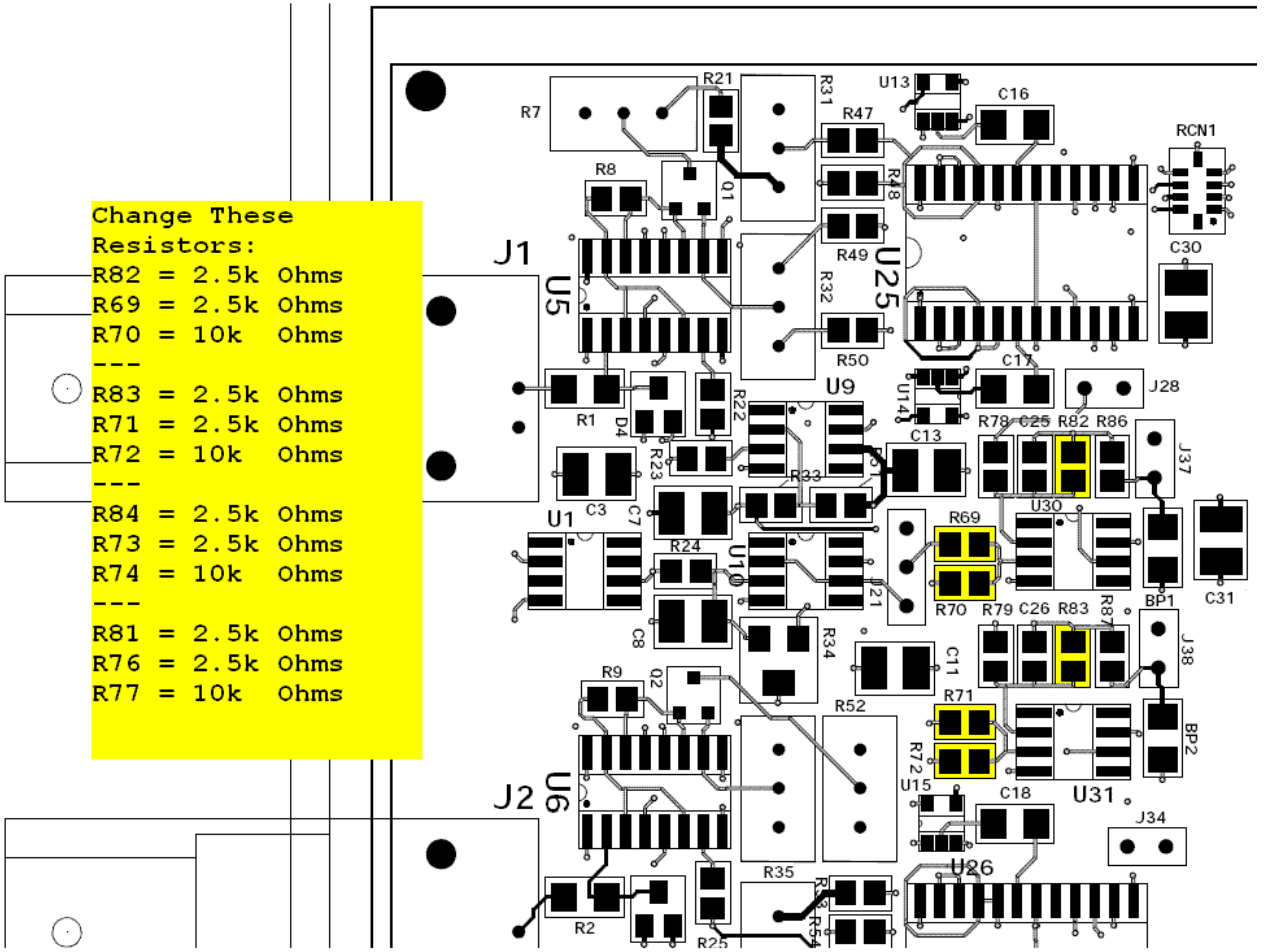
A.1 Modifications to the Module.

1. Remove all Jumpers from the DAC scaling selection pins, J13 thru J20.

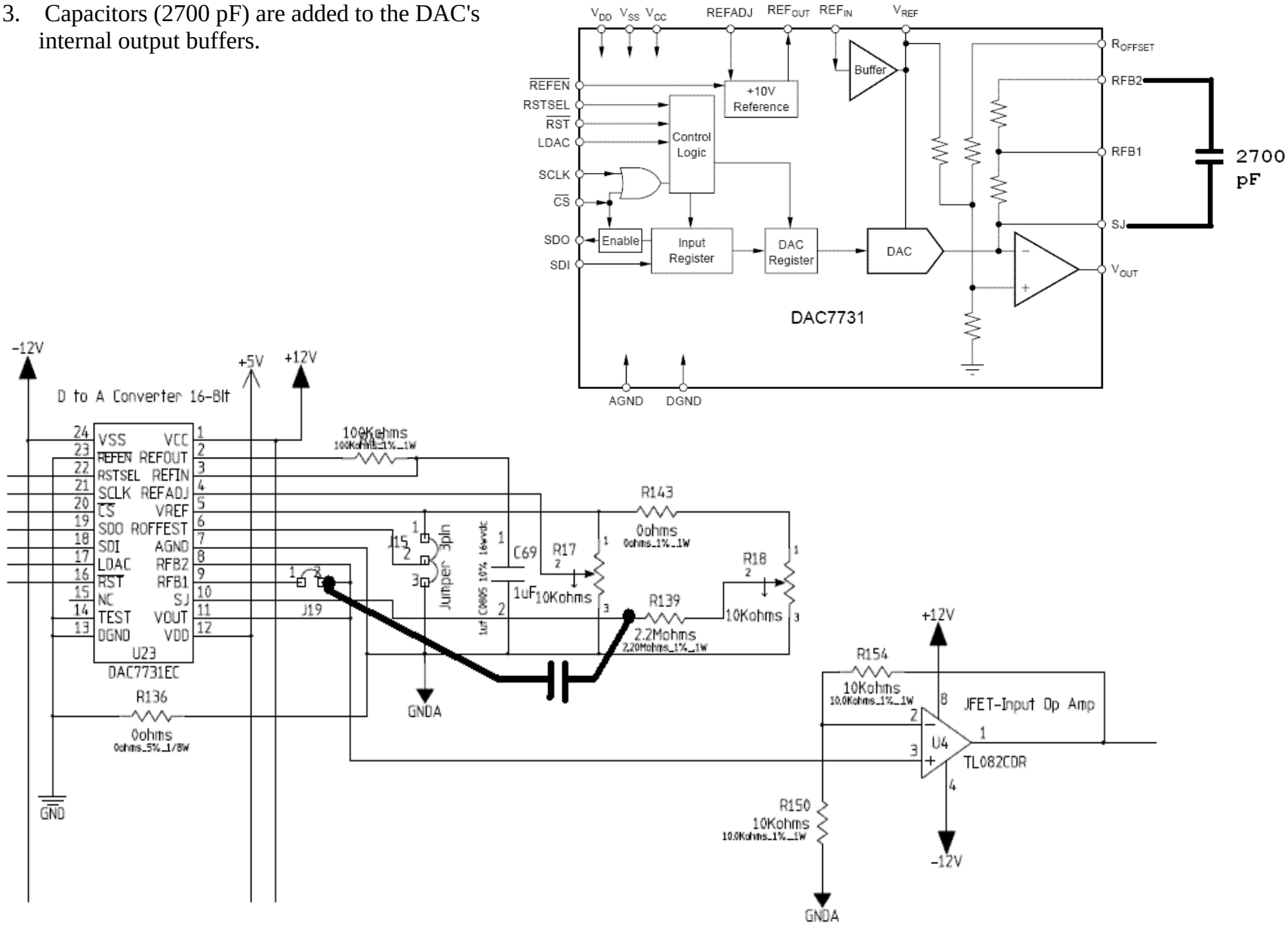


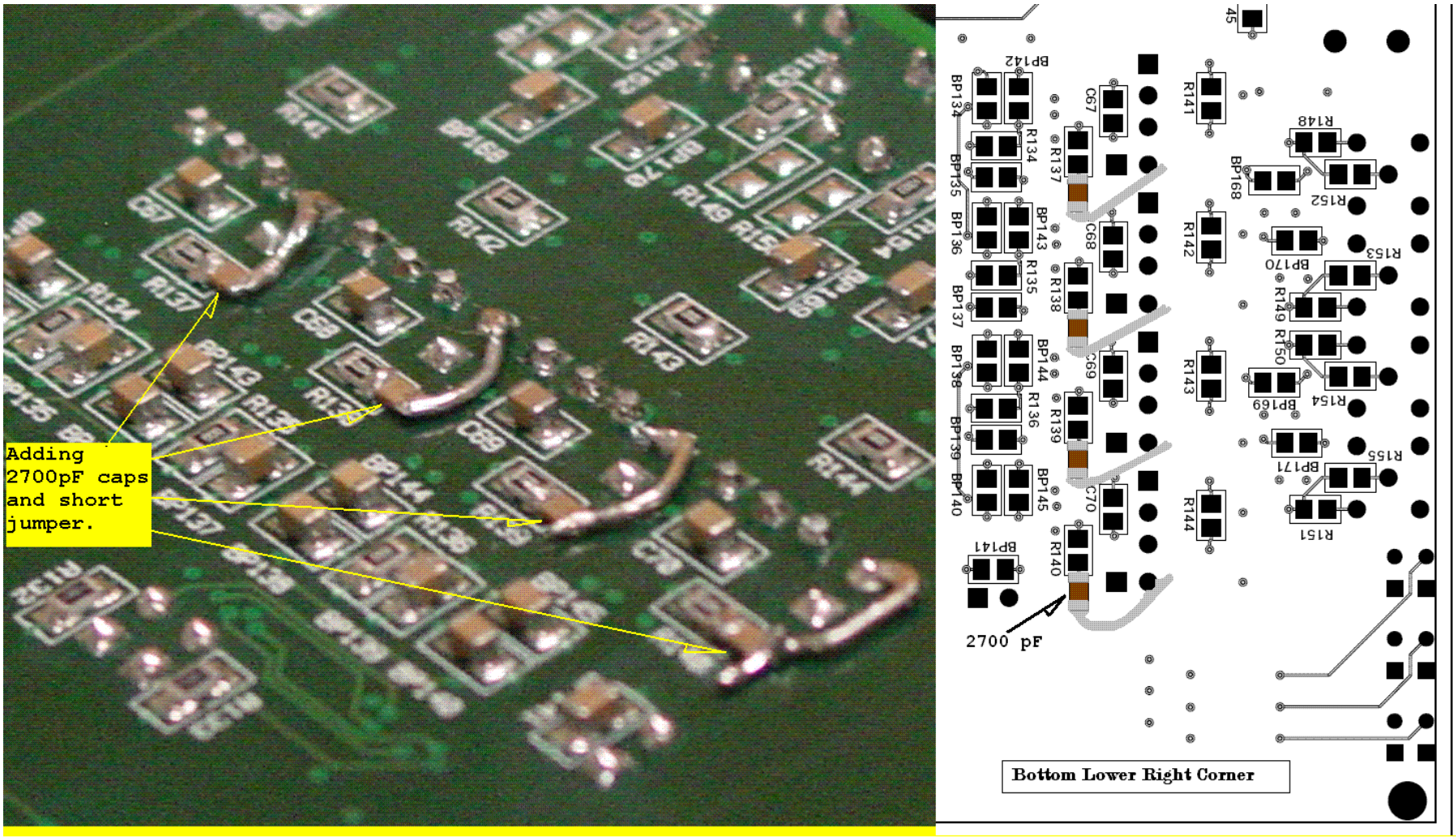
2. The ADC buffer is re-scaled to accept a +/- 10 Volt input. Inversion (x(-1)) is accomplished in the FPGA code. Change the following resistors.

** Previous values for BLM application: R81 thru R84 = 4990, R69 and R73 = 6980, R71 and R76 = 10000, (R70, R72, R74, R77) = 4990.

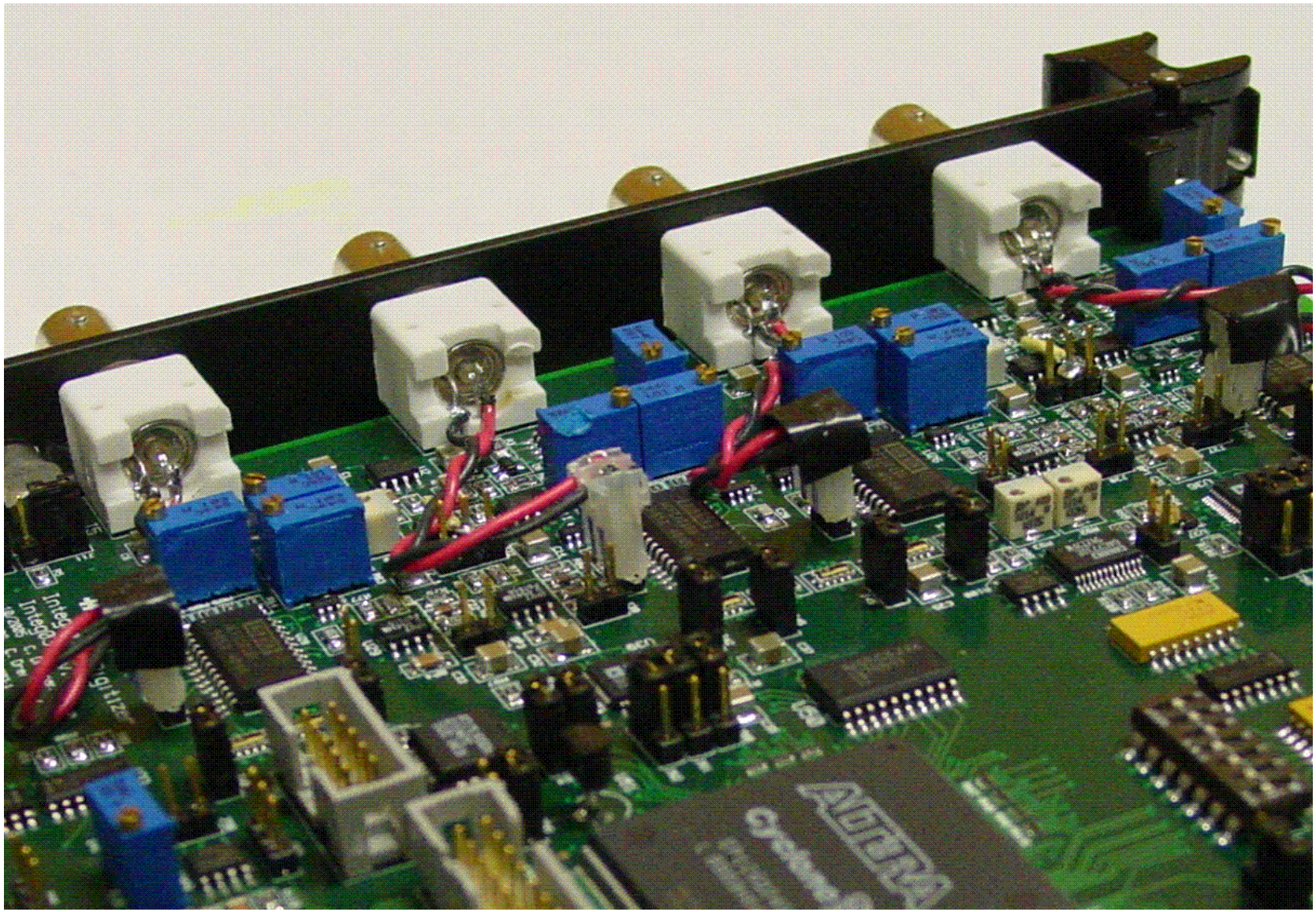


3. Capacitors (2700 pF) are added to the DAC's internal output buffers.

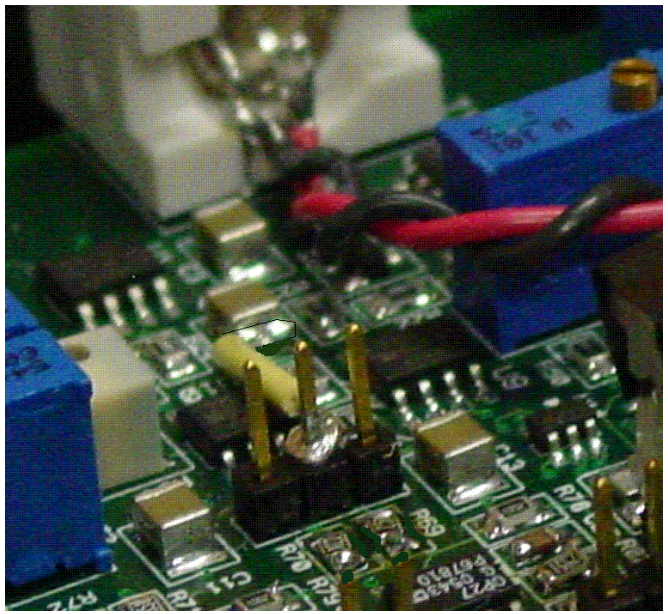


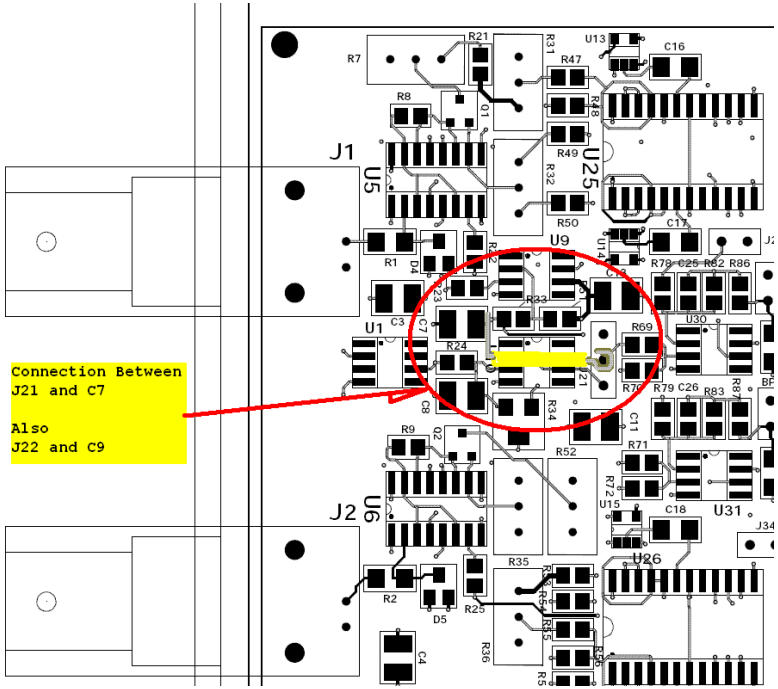

Adding 2700pF caps and short jumper.
2700 pF
Bottom Lower Right Corner

4. Jumper the input signal from the front panel BNC to the dual pin connections at J28, J29, J30, J31

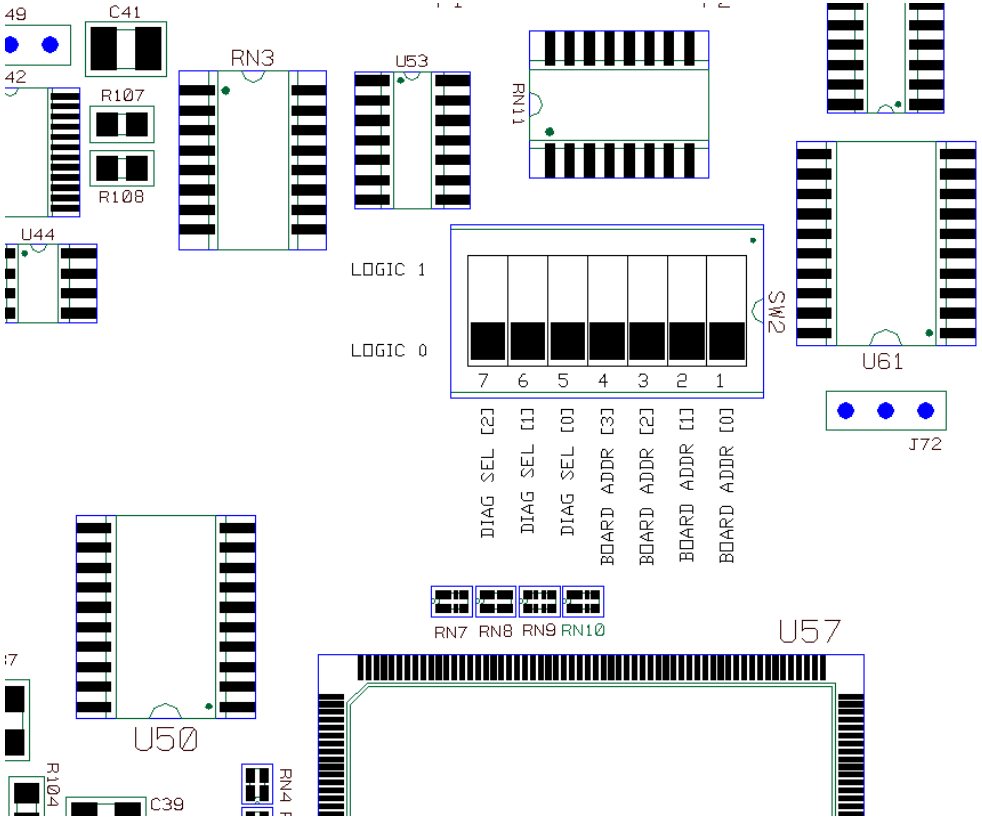


5. Add jumper between the 2.5 Volt Reference output and the bias connections to the ADC buffer amps.



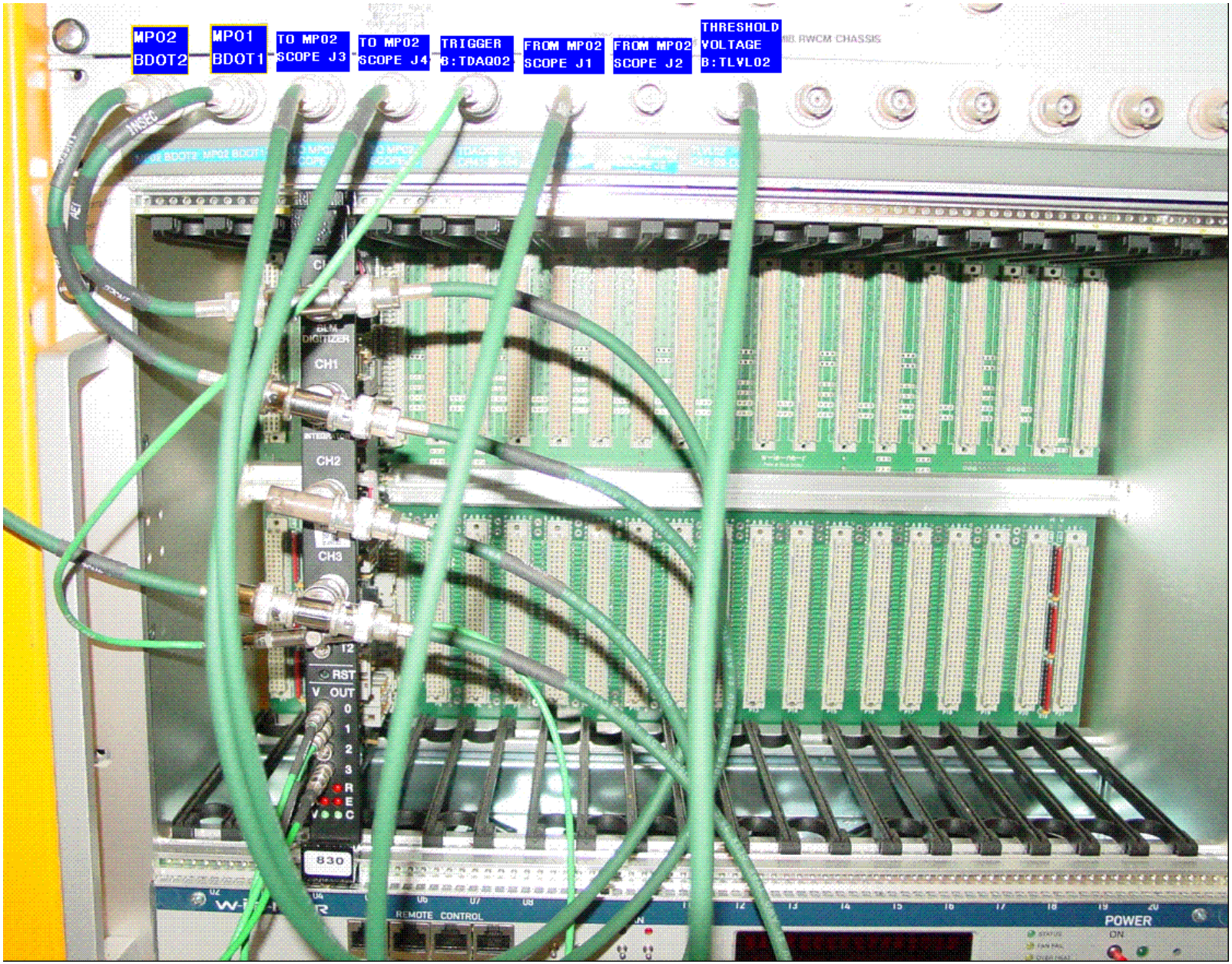


Connections between J21/C7 and J22/C9.



The Select switch for board address and diagnostic mode selection.

Switches 5, 6, and 7 should be in the up position (not shown).



Crate Hookup in rack G01-RR6-3

A.2 Bill of Material 8/12/2008

The initial requirement for parts for 5 BLM to PS modification is the following

Part Number - Part Description	Boards	Units/Brd	Total Units
RR12P2.55KDCT-ND - RES 2.55K OHM 1/10W .5% 0805 SMD	5	8	40
RR12P10.2KDCT-ND - RES 10.2K OHM 1/10W .5% 0805 SMD5	4	20	
PCC272BNCT-ND - CAP 2700PF 50V CERM CHIP 0805	5	4	20
CE100F22-2 – 2 POSITION PCB PIN CONNECTORS	5	4	20

A.3. FPGA Notes

1. SW5, SW6, SW7 are the front most dip switches near the VME J1 connector. These can be used to set different modes of operation. With [sw7,sw6,sw5] = 111 we can set the board into pulse stretcher mode.

2. Pulse stretcher mode sets the following

```
WHEN B"111" =>    -- "Reserved" (VME Bus Control, Test Enabled)
        Intg_CLK    = 20us_CLK;
        Run_Intg    = GND;
        Use_Const   = GND;
        Test_Enabled   = VCC;
        Use_Alt_ControlBus = VCC;
        INJ_TBT_TRIGx  = INJ_TBT_TRIG2;
        STDY_TBT_TRIGx = STDY_TBT_TRIG2;
        BypassInteg = VCC;  --<-- new signal
        Alt_GetADC  = 4us_CLK;    --<-- new signal

END CASE;
```

3. Added the Bypass integrator signal to control the integrator switches in the pulse stretcher mode.